Paralelización alternativa del algoritmo de Clasificación DER

Cecilia Sanz
LIDI (Laboratorio de Investigación y desarrollo en informática)
Facultad de Informática, Universidad Nacional de La Plata
La Plata (1900), Buenos Aires, Argentina
fax: 54-(0221) 4227707
e-mail: csanz@lidi.info.unlp.edu.ar

Armando De Giusti
LIDI (Laboratorio de Investigación y desarrollo en informática)
Facultad de Informática, Universidad Nacional de La Plata
La Plata (1900), Buenos Aires, Argentina
fax: 54-(0221) 4227707
e-mail: degiusti@lidi.info.unlp.edu.ar

Resumen

Se presenta una posible paralelización del método de clasificación Dynamic Evidential Reasoning utilizando imágenes hiperespectrales de regiones cultivadas.

Se han investigado y estudiado diferentes posibilidades para la paralelización del algoritmo. Una primera posibilidad analizada fue la división de los datos de la imagen entre varios procesadores cada uno realizando la clasificación determinada y un procesador maestro encargado de sincronizar y recoger los resultados parciales *(esquema master-slave con paralelismo de datos)*. En este trabajo, se presenta una paralelización alternativa a la anterior, en donde no solo se dividen los datos de la imagen a procesar entre diferentes procesadores, sino que se paraleliza el algoritmo en sí mismo utilizando un proceso para realizar la búsqueda de los soportes y uno para combinarlos *(esquema de pares cooperantes con paralelismo de datos y procesos).*

Se experimentó utilizando 4 imágenes de diferentes tamaños y cantidad de procesadores, y se analizan aquí los resultados. Se comparó la solución secuencial, con la obtenida luego de la paralelización utilizando 3, 5 y 7 procesadores de un cluster de PCs bajo Windows.

Por último, se exponen algunas conclusiones y las líneas de investigación actuales.

Palabras claves: *procesamiento paralelo, imágenes hiperespectrales, razonamiento evidencial dinámico*

Introducción

Las imágenes hiperespectrales contienen de diez a cientos de bandas espectrales conteniendo información sobre el área sensada, su clasificación resulta costosa en términos computacionales debido a su tamaño **[Jen96]**. En particular, para el algoritmo de clasificación DER que se utiliza en este caso para detectar los cultivos de las mismas, se trabaja con datos provenientes de las 10 bandas espectrales y rasgos espaciales, con lo cual cantidad de variables de entrada para el algoritmo es alto y esto aumenta aún más el tiempo de respuesta en el procesamiento **[San02]**.

Se implementó una primera solución paralela con un esquema maestro – esclavo, donde cada proceso esclavo trabajaba sobre una parte de la imagen, la procesaba y enviaba los resultados al proceso maestro, encargado de sincronizar al resto de los procesos y dar el resultado final. Esto fue implementado utilizando un cluster de PCs bajo Windows 98 y archivos compartidos para la comunicación **[San01]**.

Para la alternativa de paralelización presentada en este trabajo, se optimizó el algoritmo secuencial, y luego se implementó la paralelización mediante un proceso encargado de realizar la búsqueda de los soportes y otro proceso que combina los mismos para obtener un sólo valor de soporte y plausibilidad para cada clase para el pixel que se está clasificando **[Ped93]**. Estos procesos trabajan sobre una parte específica de la imagen, el proceso de búsqueda le envía los datos al proceso de combinación, y este último retorna los resultados parciales a un proceso coordinador o maestro, el cual da los resultados finales una vez que todos los procesos de combinación involucrados hayan finalizado. En la siguiente sección se detalla esta solución.

Análisis de la solución paralela

En la figura 1 se presenta un gráfico que permite esquematizar la arquitectura utilizada en esta solución. Como se puede apreciar, existe un proceso coordinador, y pares de procesos búsqueda - combinación de soportes que trabajan en forma sincronizada. Cada proceso se ejecuta en un procesador físico diferente.

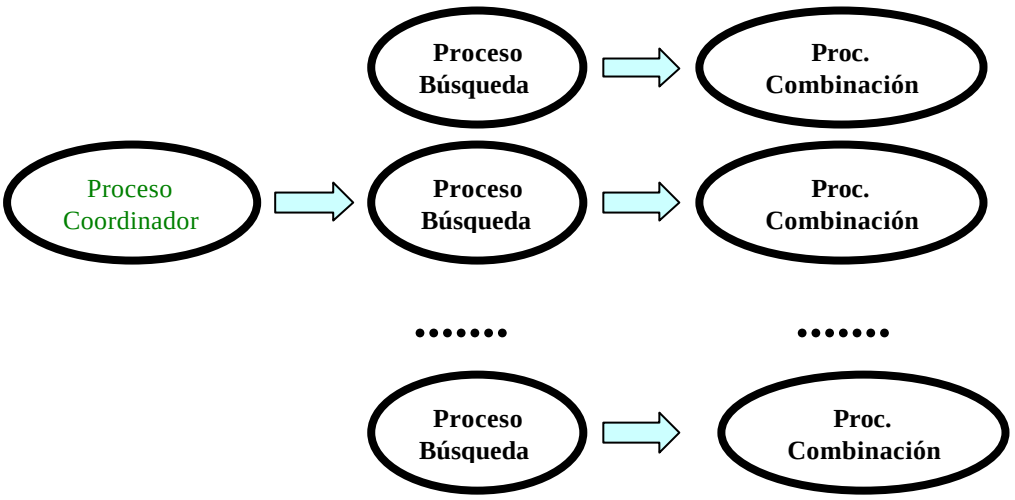
El proceso coordinador toma la imagen y la procesa inicialmente armando los vectores de datos provenientes de cada fuente (vector de rasgos), los cuales son distribuidos entre los diferentes procesos de búsqueda. Cuando el proceso de búsqueda recibe dichos datos comienza a trabajar de la siguiente manera: toma el vector de rasgos correspondiente a un pixel de la imagen, para cada valor debe buscar qué soporte le corresponde en cada una de las clases. Es decir, como en este caso se trabaja con 13 fuentes de datos y 4 clases posibles, este proceso dará como resultado una matriz de 13 x 4 para cada pixel. conteniendo el soporte que cada fuente le otorga a cada una de las clases posibles. La búsqueda de soporte se realiza sobre un arreglo de soportes (de alrededor de 2000 datos) que cada proceso de búsqueda posee localmente. El proceso de búsqueda va armando archivos con las matrices resultado, y cada archivo posee los resultados de procesar una cantidad de filas (parametrizable). Estos archivos se van enviando al proceso de combinación correspondiente, que trabaja en forma paralela a medida que recibe los datos de entrada necesarios.

El proceso de combinación toma de a una las matrices conteniendo los soportes de cada fuente para cada una de las clases, y los combina utilizando la suma ortogonal de Dempster **[Ped95]**. Luego, aplica la regla de decisión correspondiente al algoritmo DER y determina como resultado la clase a la que pertenece el pixel. Le envía una vez procesadas todas las matrices sus resultados parciales al

proceso coordinador a través de un archivo donde se resume la cantidad de pixeles encontrados para cada clase.

El proceso coordinador no da los resultados finales hasta que cada uno de los procesos de combinación le hayan enviado los resultados parciales.

Figura 1- Arquitectura del sistema de procesamiento actual-



Para esta solución se utilizó un cluster de PCs homogéneas corriendo bajo Windows 98, se realizó el experimento con los siguientes tamaños de imágenes:

Imagen	Cantidad de pixeles
Imagen 1	1160
Imagen 2	5103
Imagen 3	10404
Imagen 4	24780

Se trabajó utilizando 3, 5 y 7 procesadores, y se obtuvieron los tiempos para cada proceso involucrado y para el total del procesamiento. En la próxima sección se presentan y discuten los resultados obtenidos.

Resultados Obtenidos

En las **tablas 1-4** se presentan los tiempos de procesamiento para cada una de las pruebas realizadas. En la **figura 2** puede verse el speed up obtenido en función del número de procesadores y para cada imagen utilizada.

En la **tabla 1**, como se puede observar se presentan los tiempos de la solución secuencial y se puede ver cómo se incrementan con el crecimiento del tamaño de la imagen. En la **tabla 2**, se muestran los resultados de la solución paralela utilizando 3 procesadores, es decir, un coordinador y un par

búsqueda - combinación. Los mismos no fueron satisfactorios, dado que no mejoran los obtenidos con la solución secuencial. Esto se debe al agregado de los tiempos de comunicación, por la utilización de archivos compartidos. Sin embargo cuando se trabaja con 5 procesadores **(tabla 3)** ya pueden observarse las mejoras obtenidas, decrementando los tiempos obtenidos con 3 procesadores y la versión secuencial. Finalmente, con respecto a la utilización de 7 procesadores, puede verse que los resultados mejoraron aún más con la excepción de la primera imagen para la que se obtuvo un tiempo aproximado al procesamiento con 5 procesadores debido al tamaño de la imagen que reduce la relación *Tiempo de Procesamiento / Tiempo de Comunicación* **(tabla 4)**.

Tabla 1- ***Tiempos obtenidos con la solución secuencial***

Imágenes	***Cálculo de valor de muestra para cada fuente***	***Búsqueda de soportes para los valores de cada fuente***	***Combinación de los soportes de las fuentes***	***Tiempo total***
Imagen1	0.269 seg.	133.47 seg.	1.94 seg.	136 seg.
Imagen2	1.05 seg.	561.25 seg.	6.95 seg.	570.79 seg.
Imagen3	2.14 seg.	1210.74 seg.	15.52 seg.	1230.49 seg.
Imagen4	5.21 seg.	2723.58 seg.	34.56 seg.	2769.67 seg.

Tabla 2- ***Tiempos obtenidos con la solución utilizando 3 procesadores***

Imágenes	***Cálculo de valor de muestra para cada fuente***	***Búsqueda de soportes para los valores de cada fuente***	***Combinación de los soportes de las fuentes***	***Tiempo total***
Imagen1	0.43 seg.	135.67 seg.	123 seg.	140 seg.
Imagen2	1.47 seg.	561.83 seg.	539.69 seg.	587.64 seg.
Imagen3	2.86 seg.	1219.57 seg.	1197.76 seg.	1252seg.
Imagen4	7.25 seg.	2798.46 seg.	2750.23 seg.	2850 seg.

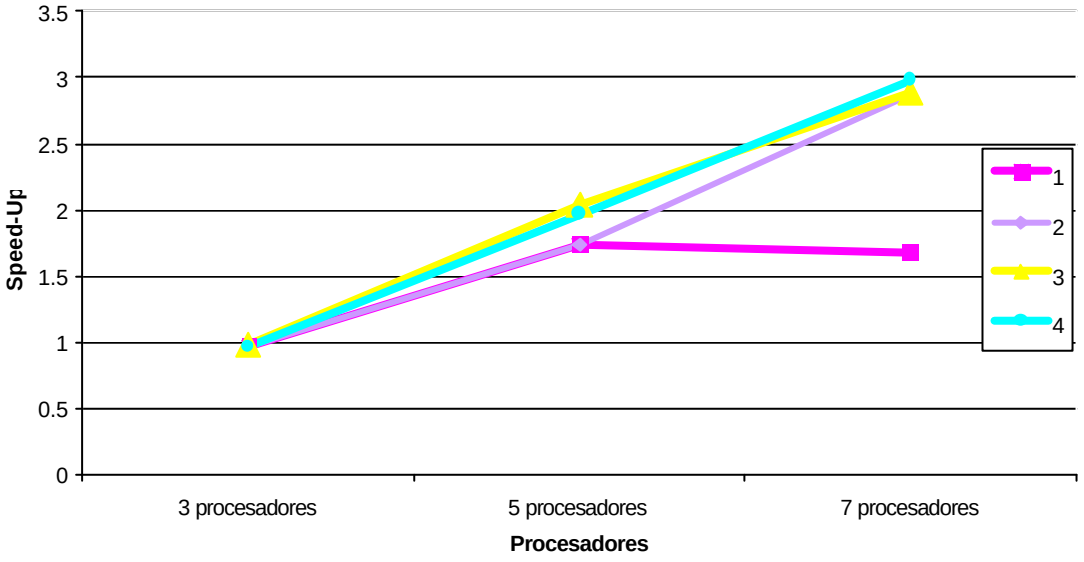
Tabla 3 - ***Tiempos obtenidos con la solución utilizando 5 procesadores***

Imágenes	***Cálculo de valor de muestra para cada fuente***	***Búsqueda de soportes para los valores de cada fuente***	***Combinación de los soportes de las fuentes***	***Tiempo total***
Imagen1	0.48 seg.	P1:69.59 seg. P2: 70.03 seg.	P1: 65.14 seg. P2: 40.21 seg.	78.05 seg.
Imagen2	1.47 seg.	P1: 322.41 seg. P2: 285.06 seg.	P1: 317.91 seg. P2: 264.09 seg.	326.37 seg.
Imagen3	2.75 seg.	P1: 607.85 seg. P2: 585.23 seg.	P1: 591.05 seg. P2: 550.52 seg.	600.45 seg.
Imagen4	7.58 seg.	P1: 1383.69 seg. P2: 1362.43 seg.	P1: 1379.29 seg. P2: 1334.48 seg.	1399.55 seg.

Tabla 4 - ***Tiempos obtenidos con la solución utilizando 7 procesadores***

Imágenes	*Cálculo de valor de muestra para cada fuente*	*Búsqueda de soportes para los valores de cada fuente*	*Combinación de los soportes de las fuentes*	*Tiempo total*
Imagen1	0.43 seg.	P1:43.89 seg. P2: 41.61 seg. P3: 46.36 seg.	P1: 35.27 seg. P2: 25 seg. P3: 39.6 seg.	80.85 seg.
Imagen2	1.43 seg.	P1: 191.14 seg. P2: 194.61 seg. P3: 177.19 seg.	P1: 189.17 seg. P2: 164.99 seg. P3: 165.54 seg.	198.50 seg.
Imagen3	2.86 seg.	P1: 407.6 seg. P2: 410.9 seg. P3: 386.73 seg.	P1: 401.12 seg. P2: 388.33 seg. P3: 382.99 seg.	424.36 seg.
Imagen4	7.97	P1: 924.94 seg. P2: 910.22 seg. P3: 904.08 seg.	P1: 922.31 seg. P2: 893.37 seg. P3: 897.81 seg.	928.08 seg.

Figura 2. Curvas de Speed Up en función de la cantidad de procesadores, para cada imagen



Discusión sobre los resultados obtenidos

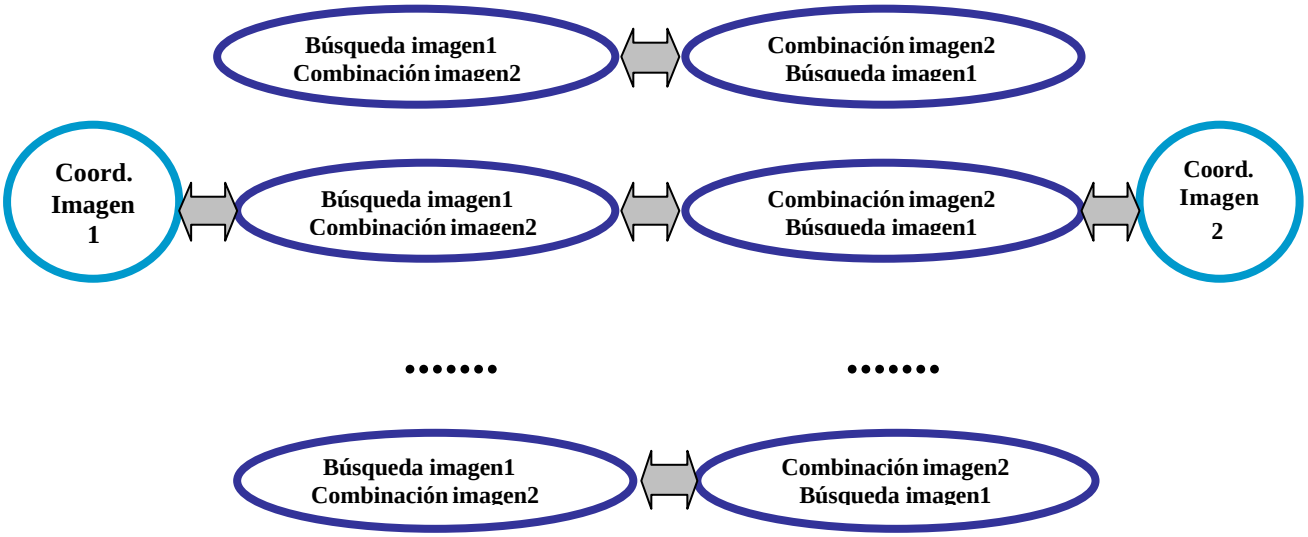
Dejando de lado la primer imagen en la que el tamaño reducido distorsiona el crecimiento prácticamente lineal del speed-up, los resultados responden a lo esperado.

Sin embargo el desbalance de carga y el incremento de las comunicaciones entre los procesadores que realizan el proceso de búsqueda con el de combinación limitan el speed up máximo alcanzable. Es interesante observar que para este nivel de complejidad de imágenes, no se mejora el speed up alcanzado con el esquema master-slave simple con 1 coordinador y n-1 procesadores esclavos que se estudió en **[San01]**.

Naturalmente la arquitectura física de procesamiento utilizada (cluster de PCs con bus Ethernet) tiene una alta relación entre el tiempo de comunicación y el tiempo de procesamiento (entre 10^6 y 10^7) lo que condiciona las ventajas de la paralelización de procesos realizada en el algoritmo original. Posiblemente para relaciones de menor orden o para un esquema de comunicaciones multi-bus el speed-up se aproximaría más al óptimo teórico **[Kum94] [Lee80]**.

Un esquema interesante que mejora el balance de carga entre los procesadores "de búsqueda" y "de combinación" es el que se muestra en la **Figura 3** donde se agrega otro proceso coordinador y se pueden tratar dos imágenes simultáneamente, alternando las funciones de búsqueda y combinación, de modo que la carga total tiende a balancearse.

Figura 3- Otra arquitectura de procesamiento planteada para desarrollar



Conclusiones y Líneas de Trabajo actuales

Actualmente se está trabajando en extender la evaluación al cluster de 16 procesadores de que dispone el LIDI, con el inconveniente de que son homogéneos "en bloques de 8", lo que requiere una corrección por el desbalance en la capacidad de procesamiento.

Por otra parte se ha estudiado la migración de algoritmos sobre la supercomputadora Clementina (SGI Origin 2000 con 40 procesadores) en la que se tiene un esquema de memoria compartida distribuida. El mayor problema es que el software de base utilizado ENVI+IDL **[Env]** no está

disponible sobre Clementina, con lo que debe migrarse el código fuente y es difícil establecer comparaciones significativas.

Bibliografía

[Env] ENVI & IDL tutorial - http://www.rsinc.com/

[Jen96] Jensen, "Introductory Digital Image Processing. A remote sensing perspective" 2da edición. Prentice Hall. 1996.

[Kum94] Kumar V., Grama A., Gupta A., Karypis G., "Introduction to Parallel Computing. Design and Analysis of Algorithms", The Benjamin/Cummings Pub. Company, Inc., 1994.

[Lee80] Lee R.B., "Empirical Results on the Speed up, Efficiency, Redundancy and Quality of Parallel Computations". Proceedings of the International Conference on Parallel Processing. 1980. Pgs. 91-96.

[Ped93] Peddle D. y Franklin S., "Classification of Permafrost Active Layer Depth from Remotely Sensed and Topographic Evidence". Remote Sensing Environment. 1993.

[Ped95] Peddle D., "Mercury: An evidential reasoning image classifier," Computers & Geosciences, vol. 21, no. 10, pp.1163-1175, 1995

[San01] Sanz C., De Giusti A., "A distributed solution for dynamic evidential reasoning applied to the classification of hyperspectral images".The 5th World Multi-Conference on Systemics, Cybernetics and Informatics, July, 2001

[San02] Sanz C., **Directores:** Jordan R., De Giusti A., "Tesis Doctoral: Razonamiento Evidencial Dinámico. Un método de clasificación aplicado al análisis de imágenes hiperespectrales". Facultad de Ciencias Exactas. UNLP. Enero 2002.